Linear SVM algorithm optimization for an EEG-based Brain-Computer Interface used by high functioning autism spectrum disorder participants

Bittencourt Villalpando, M; Maurits, N.M.

Published in:
XV Mediterranean Conference on Medical and Biological Engineering and Computing - Medicon 2019

DOI:
10.1007/978-3-030-31635-8_228

Linear SVM algorithm optimization for an EEG-based Brain-Computer Interface used by high functioning autism spectrum disorder participants

Mayra Bittencourt-Villalpando[1*][0000-0001-5248-9272] and Natasha M. Maurits[0000-0001-6231-7044]

[1] University Medical Center Groningen, University of Groningen
Department of Neurology AB51, 9700RB, Groningen, The Netherlands
*Corresponding author (e-mail: m.bittencourt.villalpando@umcg.nl)

Abstract. Machine-learning algorithms can be used for data classification on EEG-based Brain-computer interfaces (BCIs). Here, we used an algorithm based on linear support vector machine (SVM) to identify the presence of the P300 component in datasets from 15 young adult participants with autism spectrum disorder that were provided for the IFMBE Scientific Challenge 2019. We optimized the parameters and inputs for a linear SVM model throughout the ten attempts of the challenge and compared them in terms of accuracy. The highest score (mean accuracy) of 82% was achieved by a procedure that was customized per session per participant. When using a similar procedure for classification model generation and configuration of parameters for all sessions and participants, the highest score achieved was 77%. The results showed that adding data from targets from different calibration sessions from the same participants to the training dataset resulted in a significant increase in accuracy. In all attempts, the mean accuracy was above 70%, which is considered the minimum classification level for the controllability of a BCI. These are promising results for future use of BCIs as a tool for attention training in ASD participants.

Keywords: brain-computer interface (BCI), EEG, P300, Support Vector Machine (SVM)

1 Introduction

In autism spectrum disorder (ASD), deficits in the interpretation of intentions of others or other social cues are well recognized. Brain-computer interfaces (BCIs) are systems that are designed to allow the communication between the Central Nervous System (CNS) and an external device and potential use for training of social cognition skills in ASD patients has been suggested and investigated [1].

In EEG-based BCIs electrodes are placed on the scalp to measure the electrical signals from the brain. Amongst brain activity currently used for BCI operation, the P300 event-related potential (ERP) is a typical, or "naïve", brain response to a desired choice. The P300 response is evoked by the occurrence of a rare (less probable) stimulus in a sequence of stimuli and it is most frequently elicited within the framework of the "oddball paradigm". The P300 appears as a large positive deflection in the EEG signal occurring approximately 300 to 800 ms following the stimulus onset. It reflects attention related processes and it is generally observed most strongly over centroparietal brain areas [2], [3].

Studies that use machine-learning algorithms for BCI data classification have achieved impressive results for offline data analysis for healthy participants. However, the evaluation of their applicability to real life scenarios is more difficult. BCI challenges have been organized to address such problems related to BCI research and advance the knowledge in the field [4].

In this study, we used an algorithm based on linear Support Vector Machine (SVM) to discriminate segments where the P300 is present from the ones where it is not present. This algorithm was previously used by us [5] on a 4-choice BCI that was operated by healthy participants. We adapted and optimized it to identify which object a participant is paying attention to in a virtual scene, as part of the IFMBE Scientific Challenge 2019[1].

2 Methods

Determining the presence or absence of a P300 ERP can be considered a binary classification problem with a discriminant function [6]. An SVM was designed to determine the hyper-plane that maximizes the separating margin between the two classes of a binary classification between targets and non-targets (classes defined as 1 and 0, respectively).

2.1 The Dataset

The dataset with the EEG recordings of 15 participants with s=7 sessions each was provided. Data from sessions 1 to 3 were provided in Phase I and data from sessions 4 to 7 were provided in Phase II of the challenge. This dataset represents the complete EEG recordings of the feasibility clinical trial (clinical-trial ID: NCT02445625 – clinicaltrials.gov) that tested a P300-based BCI to train young adults with ASD to follow social cues [1], [7]. The EEG data provided were acquired from 8 channels (C3, Cz, C4, CPz, P3, Pz, P4 and POz). The reference was placed at the right ear and the ground electrode was placed ad AFz. Sampling rate was set at 250Hz. Data were acquired notch filtered at 50Hz and pass band filtered between 2 and 30 Hz.

The experimental design was based on a virtual scene where eight objects could be intensified. The intensification of one object is considered an event. The competition dataset was divided into train and test sets. The train set was made available with the target labels (1 out of the 8 possible events) for each block. Each block consists of a number of runs where all of the 8 possible events are intensified once in a pseudorandom order. In the dataset, data from all channels are available from -200ms to 1200ms

[1] http://www.medicon2019.org/scientific-challenge/.

relative to the onset of each event. Within each session, the train set consisted of 20 blocks with 10 runs each (with a total of 1600 events: 200 targets, 1400 non-targets) and the test set consisted of 50 blocks with a variable number of runs (Nrun) each (from 3 to 10, provided with the dataset). The challenge was to predict the labels for each block of the test set. The detailed explanation about the data acquisition process and the datasets is available elsewhere [1], [7].

2.2 IFMBE Challenge Scoring

The global score per team in the competition was defined by the attempt with the highest total accuracy and was computed using the formula:

Score (%) = #TotalCorrectLabels/#TotalLabels*100 (1),

where #TotalCorrectLabels is the total number of correct labels from all participants in sessions 4 to 7 in one attempt and #TotalLabels is the number of labels of all participants in sessions 4 to 7 in one attempt. In this challenge, #TotalLabels = 3000 (15 participants*4 sessions*50 blocks).

For each attempt of label prediction, the accuracy was provided per participant and session using the formula:

Accuracy (%) = #CorrectLabels/#Labels*100 (2)

where #CorrectLabels is the number of correct labels from one session of a participant in one attempt and #Labels is the number of labels in one session per participant in one attempt. In this challenge, #Labels = 50.

2.3 Training: model construction

The input to the SVM classifier is a feature vector formed by concatenating vectors from all eight channels. Each channel vector consists of the potentials at the time points of the ERP measured at that channel. From the training datasets provided, the data from -200ms to 0s relative to event onset (baseline) and from 1000ms to 1200ms relative to event onset were discarded. Therefore, for each channel in the dataset, datapoints from 0 to 1000 ms following each event onset were extracted, resulting in a vector length of 250 elements. The resulting channel vectors were concatenated for each event and participant, creating a single feature vector for training the classifiers. As a result, the initial training set with all 8 channels resulted in a feature vector length of l= 2000 elements (250 samples*8 channels) for each participant per event.

During the initial phase and also throughout the challenge, new parameters were defined and adjusted in an attempt to improve the classification performance. All of the parameters used are described below.

Due to the scarcity of target events in one session (1 target for 7 non-targets), two strategies were used in some of the models to increase the number of targets for the classifier:

1) Add data from target events from other sessions from the same participant: the number of training sessions (from 1 to 7) selected to build a training set was defined as k. Therefore, the total number of targets in one training set is nT=k*200.
2) Artificially increasing (x2) the number of targets in one session by pseudoran-

domly averaging the data from this session with two different pseudorandom matrices. In this case, nT=2*k*200.

The matrix $\boldsymbol{T}_s$ (200xl) consists of all feature vectors from target events in one session s. The nT feature vectors in $\boldsymbol{T}_s$ were pseudorandomly averaged for signal to noise reduction (using a randomly generated matrix $\boldsymbol{R}$(200x200) with r ones and (nT-r) zeros per column to generate a new matrix $avg\boldsymbol{T}_s$ (200xl), where r is the number of vectors selected for averaging).

The matrix $\boldsymbol{T}$ (nTxl) consists of the horizontal concatenation of the matrices $\boldsymbol{T}_s$ from all of the k sessions selected for the training set.

The matrix $\boldsymbol{NT}_s$ (1400xl) consists of all feature vectors from non-target events in one session.

The classification feature matrix $\boldsymbol{X}$ (Nxl) is the horizontal concatenation of $\boldsymbol{T}$ and $\boldsymbol{NT}_s$, where N=nT+1400. The respective class labels vector y(N) contains the class (1 for target, 0 for non-target) of the event for each feature vector in $\boldsymbol{X}$.

The SVM training algorithm ("fitcsvm", MATLAB and Statistics and Machine Learning Toolbox Release 2017b, The MathWorks, Inc., USA) returns trained SVM classifiers for binary classification based on a matrix with data from predictive features with their respective labels. $\boldsymbol{X}$ and y were used as input to the SVM training algorithm ("fitcsvm", MATLAB 2017b) using a linear kernel function, automatic kernel scaling, the default Box Constraint and standardized data predictor. The returned classifier was used for class prediction on the testing data.

2.4 Testing: label prediction

Similarly to the training data, for the testing datasets the datapoints from 0 to 1000 ms following each event onset were extracted. For each participant, the resulting data segments from all of the 8 channels were concatenated for each event, run and block, creating a testing matrix $\boldsymbol{Y}$(Nobsxl), where Nobs=8(events)*Nrun*50(blocks). For the observations of every possible event, the feature vectors from all runs in one block were averaged according to the number of runs per block, resulting in the reduced averaged testing matrix $avg\boldsymbol{Y}$(Nobs/Nrunxl). This step was done for: a) signal noise reduction, b) size reduction of the test matrix and c) simplification of the process described below to identify which event corresponds to the target.

The classification algorithm ("predict", MATLAB and Statistics and Machine Learning Toolbox Release 2017b, The MathWorks, Inc., USA) returned a matrix of scores that indicated the likelihood that a label came from a particular class (1 for target, 0 for non-target). For each observation in $avg\boldsymbol{Y}$, the predicted target corresponds to the event with the highest score.

2.5 Model parameters and Optimization procedure

The model parameters defined for attempt 1 were based on preliminary tests performed in Phase I of the challenge.

The following parameters were adjusted, as indicated in Table 1 below, throughout Phase II of the challenge:

a) The number of training sessions (k) selected for the training set: from 1 to 7.
b) Artificially increasing the number of targets as described in section 2.3: yes/no
c) The number of feature vectors r used to generate $avg\boldsymbol{T}_s$: 10 or 20.

d) The solver algorithm used as input to 'fitcsvm': Sequential Minimal Optimization (SMO, default) or L1 soft-margin minimization by quadratic programming (L1QP).
e) The weight used as input to 'fitcsvm'.
f) The 'Priors' used as input to 'fitcsvm': prior=[0.875 0.125] or auto (based on the class probability distribution from the training set, which was modified if the targets from other sessions were added as described in Section 2.3, item 1).
g) The cost used as input to 'fitcsvm': [0 1; 1 0] (auto), [0 1; 2 0] or [0 2;1 0].

Table 1. - Parameters for training of classification model per attempt

ATTEMPT	k	Training Sessions	r	Weight	nT artificially increased?	Solver algorithm	Prior	Cost
1	1	same as current testing session	10	1 for all events	No	SMO	auto	auto
2	7	all training sessions from the participant	10	7 for targets from current session 1 for targets from other sessions 1 for non-targets	No	SMO	auto	auto
3	3	"best performing" sessions (highest accuracy) from the participant	10	1 for all events	No	SMO	auto	auto
4	1	same as current testing session	10	1 for all events	No	L1QP	auto	auto
5	1	same as current testing session	20	1 for all events	No	SMO	auto	auto
6	customized			1 for all events	No	customized	auto	auto
7	7	all training sessions from the participant	10	7 for targets from current session 1 for targets from other sessions 1 for non-targets	No	SMO	prior	auto
8	7	all training sessions from the participant	10	7 for targets from current session 1 for targets from other sessions 1 for non-targets	Yes	SMO	prior	[0 1; 2 0]
9	7	all training sessions from the participant	10	7 for targets from current session 1 for targets from other sessions 1 for non-targets	Yes	SMO	prior	0 2; 1 0]
10	Selection of best performing model per participant and per session							

From attempts 1 to 5 and 7 to 9, the parameters were set according to **Table 1** to generate the classification SVM models. In attempt 5, the parameters were customized per participant and per session based on an evaluation of which adjustments previously resulted in an increase in accuracy. In the last attempt, the labels from the best performing attempt per participant and per session were selected.

3 Results

The highest score of 82% was achieved in attempt 10 by selecting the best performing model that was customized per session per participant. Boxplots of the accuracies per attempt are shown in **Fig. 1**. The average accuracies of all attempts and of all sessions (except for session 7 from participant 3 in attempt 3) were above chance level (12.5%).

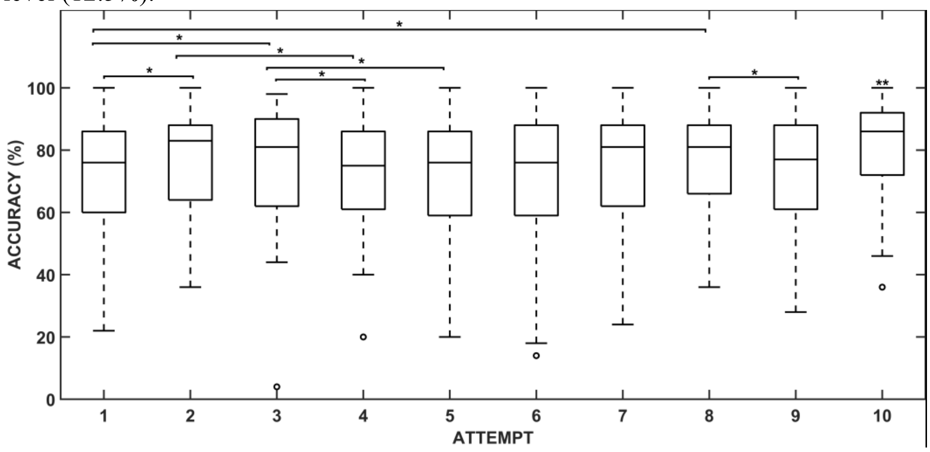

Fig. 1. – Boxplots of accuracy (%) per attempt. Asterisks indicate significant differences between attempts (Repeated measures ANOVA, *$p<0.01$, **$p<0.001$, post-hoc comparisons Bonferroni corrected)

The highest score achieved in the competition by using the same model generation procedure for all sessions for all participants was 77% for the model used in attempt 2.

Boxplots of the accuracies per participant for attempt 10, providing an overview of the variability between and within participants are given in **Fig. 2**. The highest accuracy in one session of 100% was achieved for participant 4. The accuracies were above chance level (12.5%) for all participants.

Boxplots of the accuracies per session for attempt 10 are given in **Fig. 3**. According to a repeated measures ANOVA test, there were no significant differences in accuracy across participants between sessions ($\alpha=0.05$).

The final results from the best attempt and the model selected per session and per participant are provided in the *Supplementary Material*.

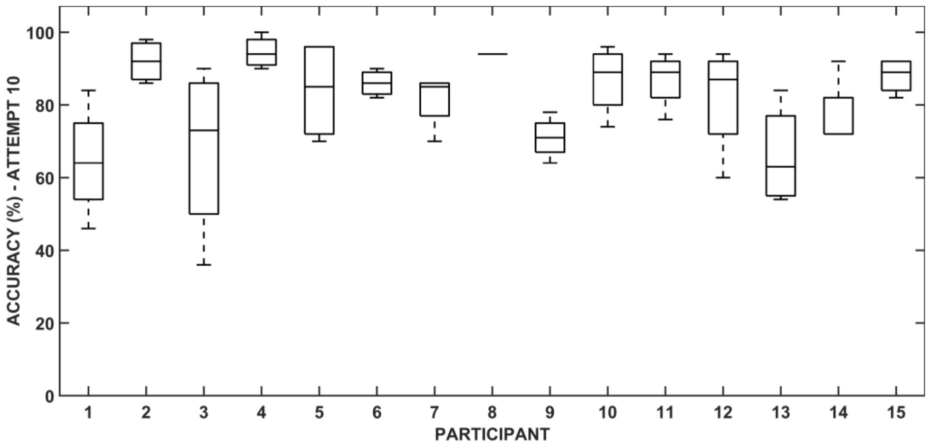

Fig. 2. – Boxplots of accuracy (%) per participant in attempt 10.

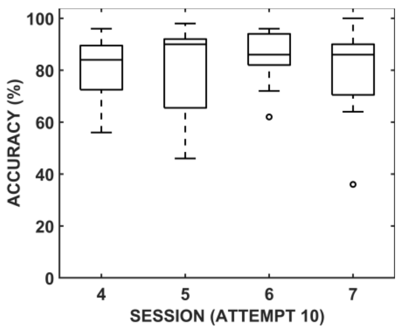

Fig. 3. – Boxplots of average accuracy (%) per session across participants for attempt 10.

4 Discussion

The main goal was to maximize the correct prediction of labels for the blocks provided in the IFMBE challenge datasets.

We optimized the parameters and inputs for a linear SVM model throughout the attempts and compared them in terms of accuracy. For the best performing attempt (attempt 10), we provided an overview of the accuracy per participant and compared the accuracy per session.

In all attempts, the average accuracy achieved was above chance level (12.5%) and the score of all ten attempts was above 70%, which is considered the minimum classification level for the controllability of a BCI [8]. Those are promising results for future use of BCIs as a tool for attention training in ASD. However, the accuracy level varied across participants and participants 1 and 13 did not achieve 70% accuracy, in any of the attempts. This might be explained by attention deficits from those specific participants given their clinical condition, although specific information about clinical condition was not made available with the dataset.

From the different optimizations that were attempted, adding data from targets from calibration sessions from the same participants resulted in a significant increase in accuracy, which is evident by the superior performance of attempt 2. Possible reasons could be (a combination of): a) the scarcity of data from targets in a single session (200 targets vs. 1400 non-targets) to train the algorithm, leading to a classification bias for the non-targets class; b) reduced performance of specific training sessions due to technical aspects (e.g. changes in the quality of the acquired signal), c) reduced attention of the participant at specific training sessions. However, it is worth mentioning that the reasons mentioned in items b) and c) should not explain the improvement in performance alone, given that no significant difference in accuracy was found between sessions.

There were no significant differences in accuracy when a) changing the solver algorithm from SMO to L1QP, b) increasing the number of events for pseudorandom averaging the targets from 10 to 20 or assigning priors (to 0.875 for non-targets and 0.125 to targets) as concluded by comparing attempt 1 to attempt 4, attempt 1 to attempt 5 and attempt 2 to attempt 7, respectively. It could not be concluded from this analysis if assigning a higher cost for misclassification of targets could increase the accuracy.

Finally, the best score was achieved in attempt 10, with a very customized solution consisting of the selection of the best performing model per session and per participant. The accuracy in attempt 10 was significantly higher than the accuracy in all other attempts. This suggests that it is very difficult to find a generic solution that fits all participants and provides stable performance across sessions. Furthermore, the computational cost involved in the different attempts differed: adding data in calibration sessions, for example, increased the accuracy but also the computational cost for training the model. This information can be relevant for the development of real-life applications. Although studies using machine learning algorithms as linear SVM have provided promising results in the literature, the optimization process remains challenging. There are several parameters that can be adjusted and their interpretation is not always straightforward. Positive results seem to depend on the extensive evaluation of the datasets and trial and error processes that are still difficult to automate and further investigation in the field is required.

REFERENCES

[1] C. Amaral *et al.*, "A feasibility clinical trial to improve social attention in Autistic Spectrum Disorder (ASD) using a brain computer interface," *Front. Neurosci.*, 2018.

[2] M. B. Pontifex, C. H. Hillman, and J. Polich, "Age, physical fitness, and attention: P3a and P3b," *Psychophysiology*, 2009.

[3] J. Polich, "Updating P300: An integrative theory of P3a and P3b," *Clinical Neurophysiology*, vol. 118, no. 10. pp. 2128–2148, 2007.

[4] B. Blankertz *et al.*, "The BCI competition III: Validating alternative approaches to actual BCI problems," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2006.

[5] M. Bittencourt-Villalpando and N. M. Maurits, "Stimuli and Feature Extraction Algorithms for Brain-Computer Interfaces: a systematic comparison," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2018.

[6] D. J. Krusienski *et al.*, "A comparison of classification techniques for the P300 Speller," *J. Neural Eng.*, vol. 3, no. 4, 2006.

[7] C. P. Amaral, M. A. Simões, S. Mouga, J. Andrade, and M. Castelo-Branco, "A novel Brain Computer Interface for classification of social joint attention in autism and comparison of 3 experimental setups: A feasibility study," *J. Neurosci. Methods*, 2017.

[8] G. Pfurtscheller *et al.*, "The hybrid BCI," *Neuroprosthetics*, vol. 4, p. 30.

SUPPLEMENTARY MATERIAL

Table 1 - The table shows the accuracy per session and per participant for all attempts. The session from attempts 1 to 9 with the highest accuracies are highlighted in gray.

PARTICIPANT	SESSION	ATTEMPT 1	ATTEMPT 2	ATTEMPT 3	ATTEMPT 4	ATTEMPT 5	ATTEMPT 6	ATTEMPT 7	ATTEMPT 8	ATTEMPT 9	ATTEMPT 10
1	**4**	60%	60%	58%	56%	66%	64%	50%	48%	54%	**66%**
1	**5**	46%	42%	44%	42%	38%	14%	36%	36%	30%	**46%**
1	**6**	54%	60%	60%	60%	58%	50%	62%	62%	58%	**62%**
1	**7**	76%	70%	78%	78%	84%	68%	66%	72%	58%	**84%**
2	**4**	86%	86%	86%	86%	86%	86%	86%	86%	86%	**86%**
2	**5**	94%	98%	96%	96%	96%	96%	96%	94%	98%	**98%**
2	**6**	88%	96%	94%	90%	90%	92%	94%	96%	92%	**96%**
2	**7**	78%	88%	86%	80%	82%	86%	88%	88%	82%	**88%**
3	**4**	88%	80%	90%	88%	86%	90%	62%	70%	62%	**90%**
3	**5**	46%	60%	58%	42%	46%	60%	64%	60%	58%	**64%**
3	**6**	78%	82%	80%	68%	72%	78%	64%	74%	66%	**82%**
3	**7**	22%	36%	4%	20%	20%	18%	24%	36%	28%	**36%**
4	**4**	96%	86%	86%	94%	92%	90%	88%	88%	88%	**96%**
4	**5**	78%	84%	86%	76%	78%	78%	84%	90%	84%	**90%**
4	**6**	88%	92%	90%	84%	84%	88%	90%	88%	90%	**92%**
4	**7**	100%	100%	98%	100%	100%	100%	100%	100%	100%	**100%**
5	**4**	52%	64%	60%	56%	54%	74%	62%	66%	52%	**74%**
5	**5**	86%	94%	96%	84%	80%	58%	86%	88%	84%	**96%**
5	**6**	92%	96%	94%	92%	92%	92%	90%	92%	84%	**96%**
5	**7**	54%	68%	46%	62%	54%	38%	62%	70%	60%	**70%**
6	**4**	74%	84%	70%	68%	66%	74%	78%	76%	74%	**84%**
6	**5**	76%	84%	88%	74%	74%	36%	88%	82%	88%	**88%**
6	**6**	66%	78%	72%	66%	70%	78%	82%	74%	74%	**82%**
6	**7**	84%	74%	84%	90%	88%	80%	84%	86%	80%	**90%**
7	**4**	62%	58%	62%	68%	70%	52%	62%	64%	68%	**70%**
7	**5**	76%	84%	70%	76%	76%	72%	70%	66%	72%	**84%**
7	**6**	68%	80%	78%	70%	60%	60%	86%	80%	82%	**86%**
7	**7**	74%	84%	82%	76%	74%	82%	82%	82%	86%	**86%**
8	**4**	84%	90%	94%	84%	88%	90%	90%	92%	90%	**94%**
8	**5**	86%	92%	92%	86%	86%	90%	90%	94%	88%	**94%**
8	**6**	92%	84%	94%	94%	90%	80%	84%	86%	84%	**94%**
8	**7**	86%	90%	86%	92%	94%	94%	92%	92%	92%	**94%**
9	**4**	76%	78%	70%	74%	76%	76%	50%	60%	54%	**78%**
9	**5**	60%	68%	70%	60%	54%	30%	62%	62%	64%	**70%**
9	**6**	52%	66%	66%	52%	54%	68%	72%	72%	68%	**72%**
9	**7**	50%	60%	56%	48%	52%	64%	52%	62%	42%	**64%**
10	**4**	64%	72%	72%	64%	62%	74%	72%	72%	72%	**74%**
10	**5**	84%	88%	90%	86%	84%	88%	90%	92%	88%	**92%**
10	**6**	68%	86%	84%	70%	68%	76%	84%	84%	86%	**86%**
10	**7**	92%	92%	92%	96%	94%	92%	92%	92%	94%	**96%**
11	**4**	78%	88%	84%	84%	80%	86%	82%	86%	84%	**88%**
11	**5**	88%	86%	90%	88%	88%	88%	88%	86%	88%	**90%**
11	**6**	86%	88%	90%	86%	86%	94%	82%	80%	76%	**94%**
11	**7**	66%	76%	70%	64%	64%	70%	66%	72%	62%	**76%**
12	**4**	78%	74%	80%	82%	84%	76%	80%	80%	78%	**84%**
12	**5**	42%	52%	54%	40%	48%	50%	58%	58%	60%	**60%**
12	**6**	82%	90%	92%	86%	84%	90%	92%	94%	90%	**94%**
12	**7**	86%	88%	88%	82%	82%	78%	90%	86%	88%	**90%**
13	**4**	24%	50%	56%	40%	42%	32%	42%	48%	32%	**56%**
13	**5**	46%	46%	54%	50%	52%	40%	46%	42%	36%	**54%**
13	**6**	74%	62%	74%	76%	74%	78%	84%	84%	84%	**84%**
13	**7**	64%	50%	62%	62%	70%	68%	62%	64%	60%	**70%**
14	**4**	54%	64%	60%	56%	56%	68%	66%	72%	68%	**72%**
14	**5**	70%	92%	82%	64%	66%	34%	66%	86%	70%	**92%**
14	**6**	52%	64%	56%	52%	50%	28%	70%	72%	66%	**72%**
14	**7**	50%	58%	58%	50%	56%	72%	54%	56%	46%	**72%**
15	**4**	82%	88%	90%	82%	82%	90%	92%	92%	90%	**92%**
15	**5**	92%	90%	88%	90%	88%	46%	88%	88%	90%	**92%**
15	**6**	68%	74%	78%	74%	78%	40%	76%	82%	76%	**82%**
15	**7**	72%	84%	82%	74%	76%	84%	80%	82%	86%	**86%**
AVERAGE		72%	77%	76%	72%	72%	70%	75%	76%	73%	**82%**